

**THE ENHANCED
SOFTWARE LIFE CYCLE
SUPPORT ENVIRONMENT (ProSLCSE):
AUTOMATION FOR ENTERPRISE AND PROCESS MODELING**

James R. Milligan
Rome Laboratory
ProSLCSE Program Monitor
RL/C3CB
Griffiss AFB, NY 13441-5700
315-330-2054

James E. Dutton
ISSI
ProSLCSE Program Manager
9430 Research Blvd.
Austin, TX 78729-6543
512-338-5729

1.0 INTRODUCTION

Rome Laboratory, Griffiss AFB, New York, and Electronic Systems Center (ESC), Hanscom AFB, Massachusetts, are joint sponsors of a five year program entitled the *Software Life Cycle Support Environment (SLCSE) Enhancements and Demonstration Program* to develop and support a state-of-the-art Software Engineering Environment (SEE) product for software development and post-deployment support.

SLCSE, pronounced "slice", exists today as an advanced development prototype (completed in late 1989) for the product currently being developed, called *ProSLCSE*, by International Software Systems, Inc. (ISSI), Austin, Texas.

During the year following the completion of SLCSE, the prototype environment was beta tested at three USAF AFLC Air Logistics Centers. In addition, SLCSE underwent several independent evaluations. As a result of the beta tests and evaluations, there was overwhelming agreement that the SLCSE concept was a sound one that could significantly impact the Air Force software process in a very positive way. The beta tests and evaluations, however, also revealed several usability and performance issues that need resolution before SLCSE can be fielded for widespread operational use. ProSLCSE is being developed with these very issues in mind.

To increase performance, usability and portability, the ProSLCSE architecture will migrate from the proprietary VAX/VMS based SLCSE architecture to one which supports a network of heterogeneous POSIX workstations. The ProSLCSE user interface will migrate from the SLCSE character-oriented display (e.g., for VT-100 type terminals) to the X-Windows/MOTIF presentation style. The ProSLCSE repository will migrate toward an ECMA PCTE compliant, client-server architecture. Tool improvements will include, among others, CALS-compliant DOD-STD-2167A automatic document generation, and enhanced life cycle traceability and impact analysis capabilities. Undoubtedly, the most significant improvement in ProSLCSE over SLCSE will be in its enterprise and process management capabilities.

The first product release of ProSLCSE is expected in December 1993. Additional product releases/upgrades will continue during and subsequent to the five year contract (which began August 1991) that will produce a commercial-quality product for use by Government organizations,

Government contractors, and industry. ISSI will also provide comprehensive customer support services such as product installation, user-customized training, on-site software support, mission-specific demonstrations, application consulting, tool/system integration with ProSLCSE, and/or the development of specifically desired ProSLCSE capabilities.

2.0 ProSLCSE

ProSLCSE provides a computer-based *framework* that is used to instantiate *environments* that are tailored to accommodate the specific needs (in terms of process, users, data, and tools) of any software development/support *project*. A ProSLCSE environment supports a total life cycle concept where an integrated toolset is applied during each software development phase (concept exploration, demonstration and validation, and engineering and manufacturing development) and is later transitioned, along with an associated repository containing all accumulated information, to the Post-Deployment Software Support (PDSS) phase. The intent of this concept (as shown in Figure 1) is not only to increase productivity and product quality during the developmental phases, but also to improve the supportability of the product by making available to Software Development Support Activities (SDSAs) all development data and a highly effective PDSS toolset.

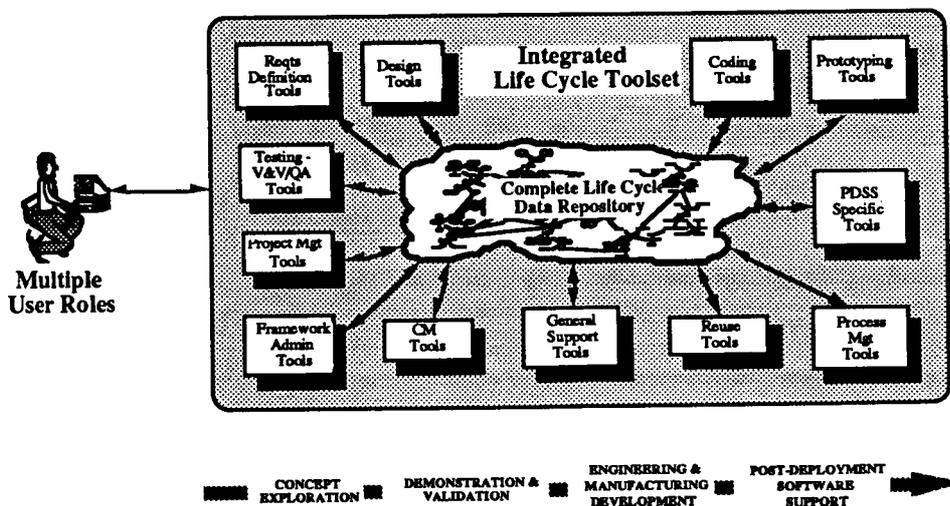


Figure 1 - ProSLCSE: Software System Support from "Cradle To Grave"

2.1 A Process-Oriented Framework

Regardless of the life cycle phase, the operational concept of ProSLCSE always centers around the *process* for a project in which users of the system have jobs to do. The nature of a user's job at any particular time is characterized by the user's *role* which defines the scope of the work that can

be performed by that user (e.g., project management or programming). The roles of users are dependent on the *active* tasks that are currently being performed. User roles, in turn, determine the tools and data that are accessible to users in order to perform those tasks. The user can choose from any one of their *ready* tasks and make it an active task, but cannot perform a *pending* task until all data upon which it is dependent becomes available.

The entire set of tasks, data stores, data flows (dependencies) between tasks/data stores, roles associated with tasks, and personnel-to-task assignments defines the process to be followed for the project in which the user plays one or more roles. ProSLCSE provides the guidance necessary to ensure that users perform task assignments in such a way that does not violate a project's process definition. Figure 2 illustrates a simplified example of this.

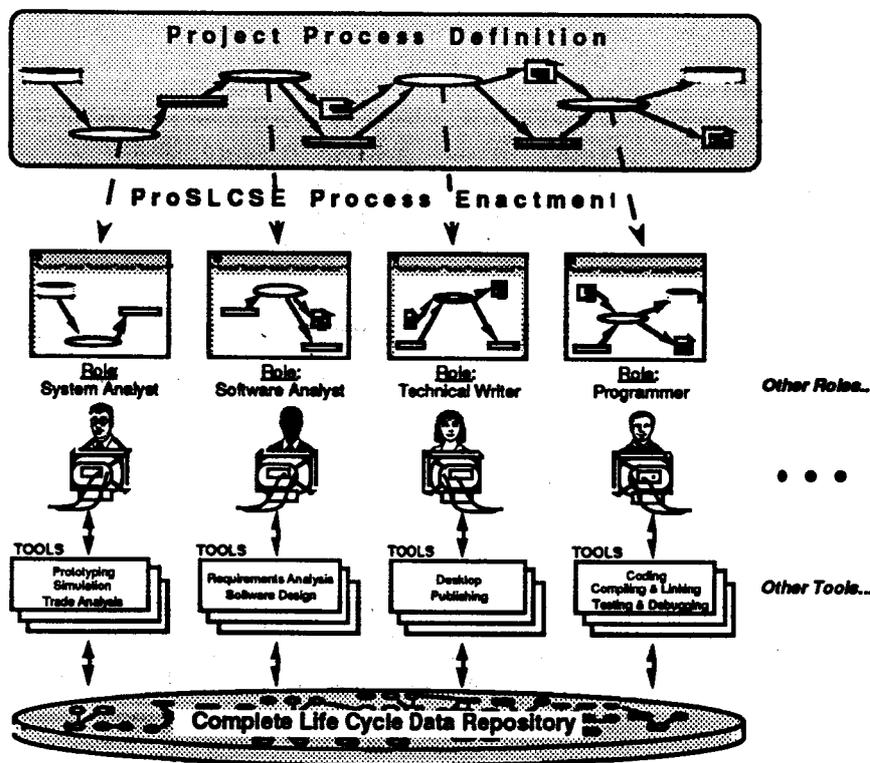


Figure 2 - ProSLCSE: A Process-Oriented Environment Framework

2.1.1 Enterprise Modeling

Enterprise modeling can be defined as the development of the following three submodels:

- Infrastructure Model
- Information Model
- Process Model

While each of these models is conceptually distinct from the other, primarily because it is easier to develop each model independently, the three are far from being physically separate. In fact, ProSLCSE will provide the automated tools necessary to define and tightly integrate all three models within a common life cycle data repository (as discussed in section 2.1.2).

The *Enterprise* and *Process* modeling techniques and representations introduced in this paper comprise a methodology known as *FirstEP* (pronounced "first-step"). In *FirstEP*, each model is defined in terms of a declarative, graphical language [reference to JD's paper] in order to attain the proper mix of expressiveness and ease of use that will enable non-programmers to define complex, real-world situations. The concepts behind each model are now discussed.

2.1.1.1 Infrastructure Model

The infrastructure model describes the people who work in an organization, the tools and other resources they use to perform their jobs, the formal and informal groups to which they belong, their relationships with one another, including reporting relationships, and the communication paths that exist among groups. As shown in Figure 3, infrastructure model concepts include resource, person, tool, machine, location, and group.

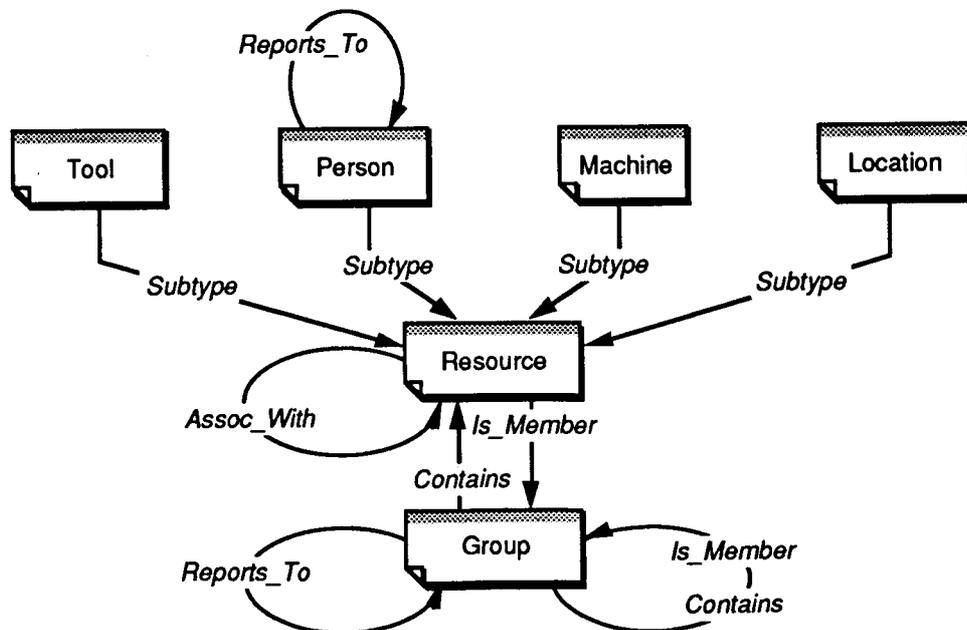


Figure 3 - Infrastructure Model Components

2.1.1.2 Information Model

The information model describes the types of objects, such as documents, code, forms, and messages, that are manipulated and exchanged by resources within the organization's infrastructure, as well as the relationships among those object types and the allowed kinds of groupings of objects. As shown in Figure 4, the information model includes the concepts of object, link, object type, link type, attribute, contents, and composite.

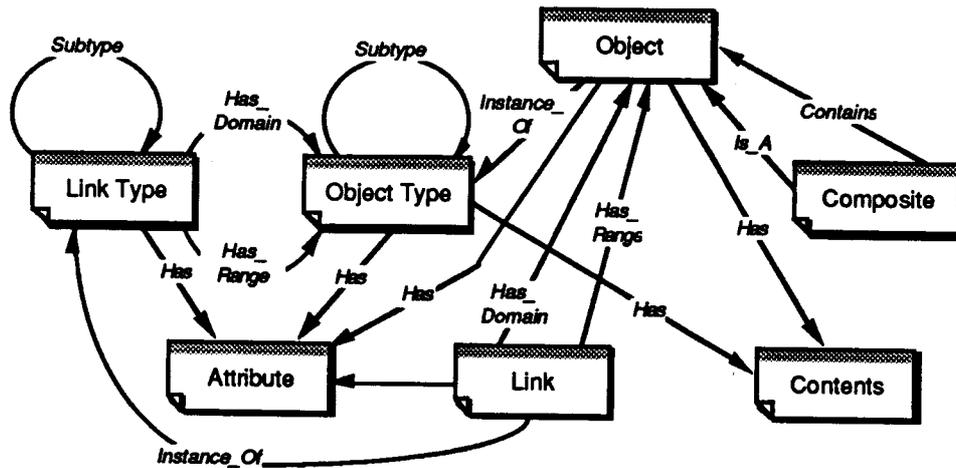


Figure 4 - Information Model Components

2.1.1.3 Process Model

The process model describes tasks to be performed, the resources required to perform those tasks, the objects required and produced by each task, and the work breakdown structure in terms of how tasks are refined (decomposed) into lower-level tasks. As shown in Figure 5, the process model includes the concepts of project, process, task, and product.

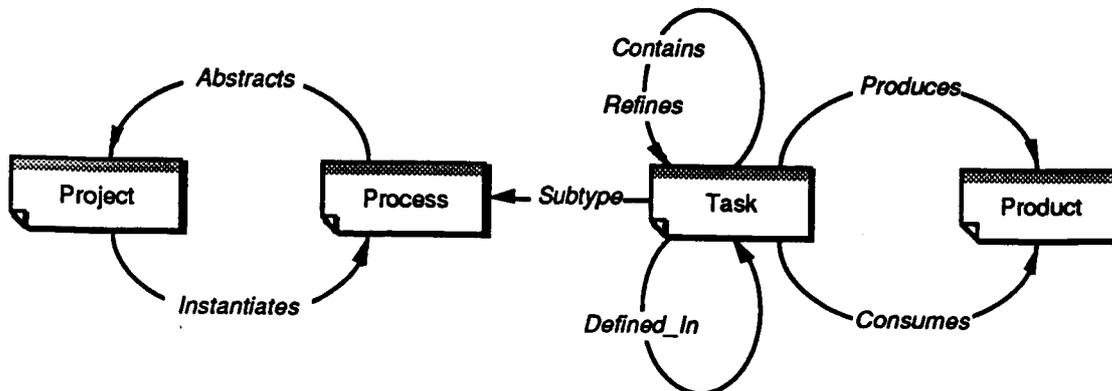


Figure 5 - Process Model Components

2.1.1.4 Relationships Between Models

Although each model can at first be defined independently, the establishment of a complete enterprise model comes about only when the relationships between the three models are instantiated. Figure 6 shows some of the relationships between components of each model. The seamless integration of the infrastructure, information, and process models is the key to establishing a controlled environment in which work flow efficiency and quality are optimized.

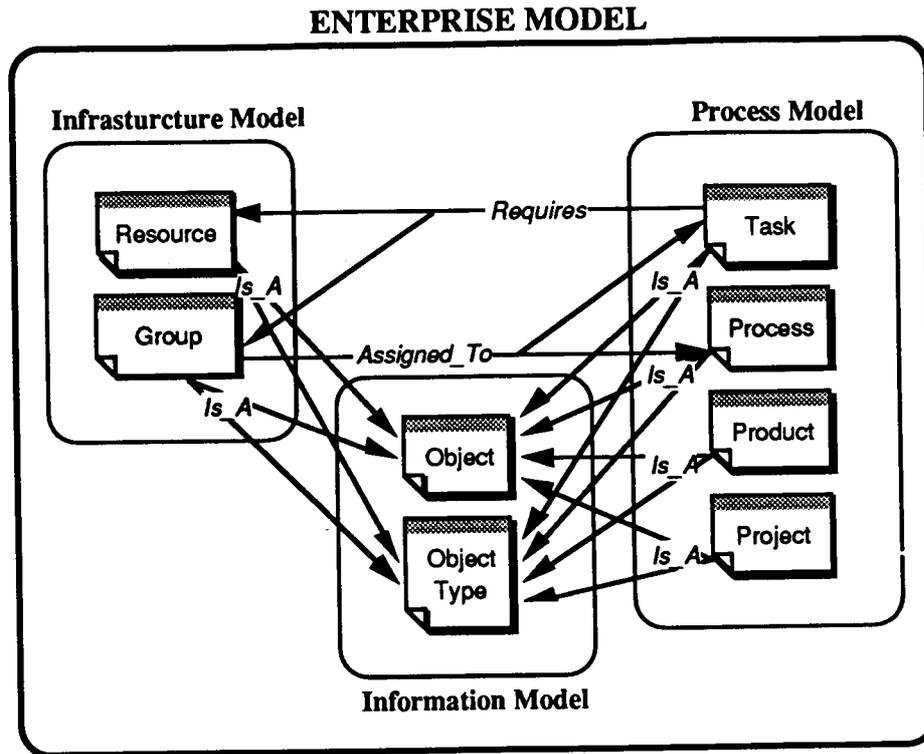


Figure 6 - Inter-Model Component Relationships

2.1.2 Enterprise Modeling and Process Enactment In ProSLCSE

A visual process modeling language called *ProVision* will be used in ProSLCSE to facilitate the definition of project processes. Using a ProVision editor, it will be possible to model tasks and communication paths in the form of a directed graph where tasks are represented as nodes, and communication paths (data flow) are represented as edges that connect tasks. Different kinds of communication (e.g., documents, electronic mail, computer input/output, verbal, etc.) are represented by smaller icons along the communication paths.

Similarly, ProSLCSE will provide a visual data definition language editor to define a project's information model, and a visual infrastructure language editor to define an organization's infrastructure model.

Once an environment is instantiated from the enterprise model defined, all model information is stored in a common ECMA PCTE repository. Based on repository information, ProSLCSE will control user work flow, while at the same time, will ease the difficulty of proper task execution by providing a structured work environment to the user.

3.0 CONCLUSION

In this paper, we have introduced a comprehensive method for enterprise modeling that addresses the three important aspects of how an organization goes about its business. FirstEP includes infrastructure modeling, information modeling, and process modeling notations that are intended to be easy to learn and use. The notations stress the use of straightforward visual languages that are intuitive, syntactically simple, and semantically rich.

ProSLCSE will be developed with automated tools and services to facilitate enterprise modeling and process enactment. In the spirit of FirstEP, ProSLCSE tools will also be seductively easy to use.

Achieving fully managed, optimized software development and support processes will be long and arduous for most software organizations, and many serious problems will have to be solved along the way. ProSLCSE will provide the ability to document, communicate, and modify existing processes, which is the necessary first step.

4.0 REFERENCES

1. Dutton, James, "FirstEP: A Common Sense Approach to Enterprise and Process Modeling", ISSI-A92A02001, International Software Systems, Inc., Austin, Texas, June, 1992.
2. Milligan, James, "The Enhanced Software Life Cycle Support Environment: SEE It Today (Tomorrow May Be Too Late)", 3-46, Fourth International Conference on Strategic Software Systems, Huntsville, Alabama, March, 1992.
3. "Proposal for the Software Life Cycle Support Environment (SLCSE) Enhancements and Demonstration Program", ISSI-R9102003, International Software Systems, Inc., Austin, Texas, May, 1991.
4. "Software Design Document for the Enhanced Software Life Cycle Support Environment CSCI of the E-SLCSE System", ISSI-B91A00008A, International Software Systems, Inc., Austin, Texas, June, 1992.
5. "Software Requirements Specification for the E-SLCSE CSCI of the Software Life Cycle Support Environment (SLCSE) Enhancements and Demonstration

Program", ISSI-B91A00007A, International Software Systems, Inc., Austin, Texas, February, 1992.

6. "Statement of Work for the Software Life Cycle Support Environment (SLCSE) Enhancements and Demonstration Program", Rome Laboratory, Griffiss Air Force Base, New York, January, 1991.